Artificial Intelligence

Artificial Intelligence (AI) may be defined as the branch of computer science that is concerned with the automation of intelligent behavior.

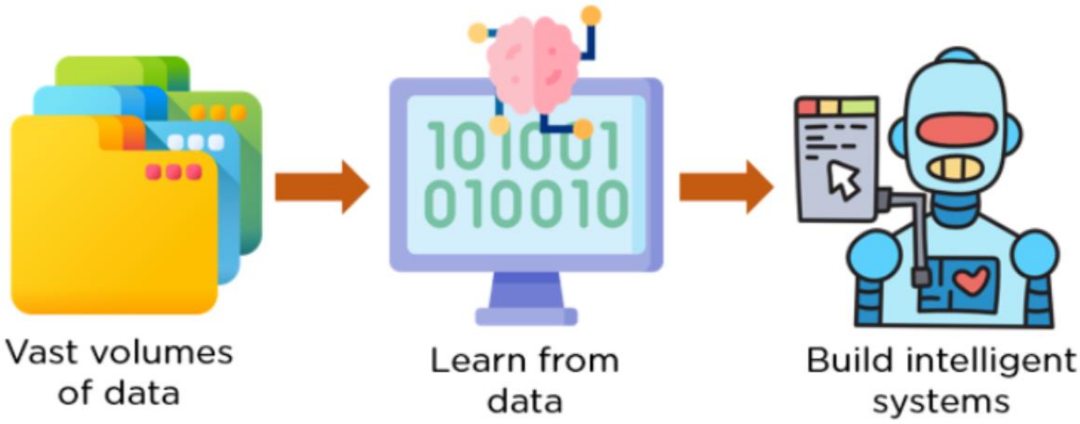
Artificial intelligence (AI) is a set of technologies that enable computers to perform a variety of advanced functions, including the ability to see, understand and translate spoken and written language, analyze data, make decisions, and more.

Artificial Intelligence (AI) works by simulating human intelligence through the use of algorithms, data, and computational power. The goal is to enable machines or software to perform tasks that typically require human intelligence, such as learning, reasoning, problem-solving, perception, and language understanding.

Some examples of AI. Your smartphone uses AI, as do services like, chatbots, social media websites, and much more.

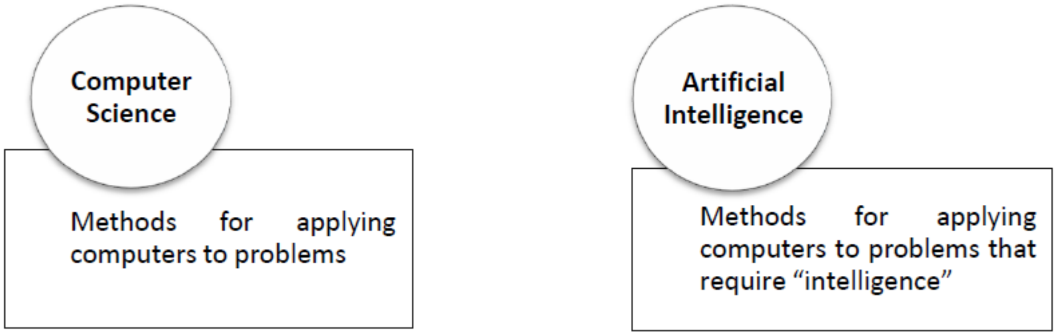
An AI tool is a software application that uses artificial intelligence algorithms to perform specific tasks and solve problems. AI tools can be used in a variety of industries, from healthcare and finance to marketing and education, analyze data, and improve decision-making.

Artificial Intelligence is the process of building intelligent machines from vast volumes of data. Systems learn from past learning and experiences and perform human-like tasks. It enhances the speed, precision, and effectiveness. AI uses complex algorithms and methods to build machines that can make decisions on their own.



Example of the human AI is Sophia, a socially intelligent humanoid robot, was developed by the Hong Kong-based engineering and robotics company Hanson Robotics. Compared to previous robots, Sophia is very advanced and can mimic human gestures as well as hold a simple conversation.

The two most fundamental concerns of AI researchers are knowledge representation and search. The first of these, which is also called Natural Language Processing (NLP).



Artificial Intelligence (AI): - To build programs that can perform intelligence tasks.

- Languages and Environments for AI

Some of the most important by-products of artificial intelligence research have been advances in programming languages and software development environments. For a number of reasons, including the size of many AI application programs, the tendency of search algorithms to generate huge spaces, and the difficulty of predicting the behavior of heuristically driven programs, AI programmers have been forced to develop a powerful set of programming methodologies. Programming environments include knowledge- structuring techniques such as object-oriented programming.

As for languages not used in AI, they are Basic, Pascal, and Fortran. COBOL, Ada.

- Characteristics of Artificial Intelligence

1- Learning Capability

One of the main characteristics of AI is its capacity to be able to draw lessons from the data. Machine learning algorithms are used to process information, identify patterns, and make decisions.

2- Problem-Solving

AI is a master at solving difficult issues, including those humans are unable to solve. AI employs algorithms and data to analyze the problems and come up with efficient solutions. Often, it is faster than any human would.

3- Reasoning

Thinking is a different key characteristic that AI has. AI is able to make rational decisions in reliance on the data it holds, utilizing rules and logic in order to come to a conclusion.

4- Perception

AI's ability to sense the environment around it is a different characteristic that makes AI stand out. AI uses sensors and data to comprehend the environment around it, much like the way humans utilize their senses.

5- Adaptability

It is the most important characteristic of AI applications. AI systems are constantly evolving and adaptable to the latest information or changes in their surroundings without needing to reprogram.

6- Data Handling

AI's capacity to process vast quantities of data is an amazing characteristic. AI is able to store, process data, and analyze it with a speed that humans cannot.

7- Natural Language Processing (NLP)

Natural Language Processing (NLP) lets AI comprehend and communicate with humans in a way that makes it much easier to communicate with machines.

8- Efficiency

One of the attractive features of AI applications is that they can improve effectiveness. AI is able to complete tasks faster and more precisely than human beings, thereby saving time and energy.

9- Decision-Making

The ability of AI to make choices is crucial to many sectors. AI makes use of data analysis as well as pattern recognition in order to come up with accurate and informed decisions.

AI Application

Artificial Intelligence has various applications in today's society. It is becoming essential for today's time because it can solve complex problems with an efficient way in multiple industries, such as Healthcare, finance, education, etc. AI is making our daily life more comfortable and fast.

Following are some sectors which have the application of Artificial Intelligence:

- Data, Information, and Knowledge

1- Data : is the fundamental, indivisible objects in the application.
2- Information: is the implicit functional association between data in the application.
3- Knowledge: is the explicit functional association between items of information and/or data in the application.

Inference: which is a method of logical resulting in the scripts of facts and rule by using heuristics.

Heuristics: which is a mean of judge on objects by expert experimentally.

Search Methods:

Problem solving:

There are three major steps required in building systems that solve particular problem

1. Problem definition: detailed specification of inputs and what constitutes an acceptable solution.
2. Problem analysis, Knowledge representation.
3. Choose the best technique and apply it to the particular problem.

The component of intelligent problem solving is search. Humans generally consider a number of alternative strategies on their way to solving a problem. A chess player typically reviews alternative moves, selecting the "best" according to criteria such as the opponent's possible responses or the degree to which various moves support some global game strategy. A player also considers short-term gain (such as taking the opponent's queen), opportunities to sacrifice a piece for positional advantage, or conjectures concerning the opponent's psychological makeup and level of skill. This aspect of intelligent behavior underlies the problem-solving technique of state space search.

-Structures for state space

Basic definitions in graph theory

A graph is a set of nodes connected by arcs if there is a direction associated with the graph then the graph is called directed.

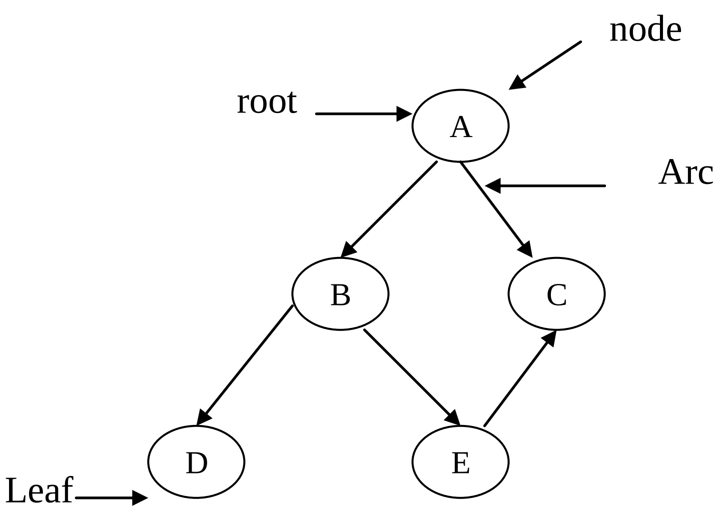
A root is a node with no parent. A rooted graph has a unique node called the root. such that there is a path from the root to all nodes within the graph. In drawing a rooted graph, the root is usually drawn at the top of the page, above the other nodes.

A tip or leaf is a node with no child

A path is an ordered sequence of nodes.

A cyclic path is a path that contain the same state more than one.

A tree is a graph in which there is a unique path between every pair of nodes (the paths in a tree, therefore, contain no cycles). It's a graph in which tow nodes have at most one path between them. Trees often have roots, in which case they are usually drawn with the root at the top, like a rooted graph. Because each node in a tree has only one path of access from any other node, it is impossible for a path to loop or cycle continuously through a sequence of nodes. Terms used to describe relationships between nodes include parent, child, sibling. The children of a node are called siblings, two nodes in a tree or graph are said to be connected if there is a path between them.



Graph

A graph consists of:

A set of nodes $N_1, N_2, N_3, \ldots, N_n, \ldots$, which need not be finite.

A set of arcs that connect pairs of nodes.
(Arcs are often described as an ordered pair of pair of nodes; i.e. the arcs (N_3,N_4) connects node N_3 to N_4).
A directed graph has an indicated direction for traversing each arc. For example, a directed graph might have (N_3,N_4**)** as an arc but not (N_4,N_3). This would indicate that a path through the graph could go from N_3 to N_4 but not from N_4 to N_3.

State Space represented of problems

In the state space represented of a problem, the nodes of a graph corresponding to partial problem solution states and the arcs corresponding to steps in a problem-solving process. State space search characterizes problem solving as the process of finding a solution path from the start state to goal.
To build a program for solving any problem, it would first be necessary to specify the following:

- The Start state for the problem.
- The Goal state, the state represent a solution to the problem.
- The rules that define the legal moves through the stat space.

State Space Search

A state space is represented by a four-tuple [N,A,S,G], where :
N is the set of nodes or states of the graph, these corresponding to the states in a problem-solving process.
A is the set of Arcs (or link) between nodes. These corresponding to the steps in a problem-solving process.
S a nonempty subset of N, contains the start state(S) of the problem.
G a nonempty subset of N, contains the goal state(G) of the problem.

A solution path is a path through this graph from a node in S to a node in G.

The state space representation provides a mean of determining the complexity of the problem.

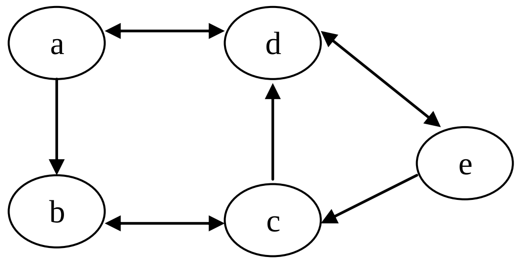
Start state : a

Goal state : e

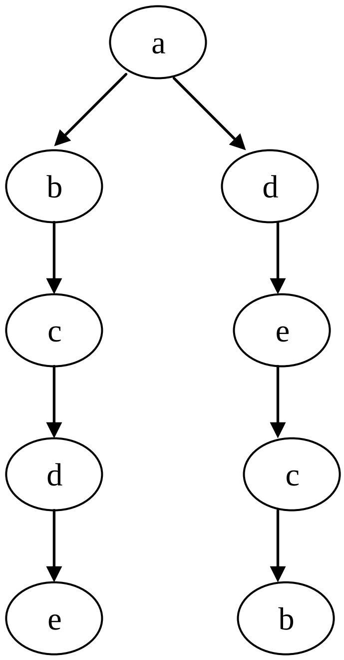
Path1 : a-d-e

Path2 : a-b-c-d-e

cycle



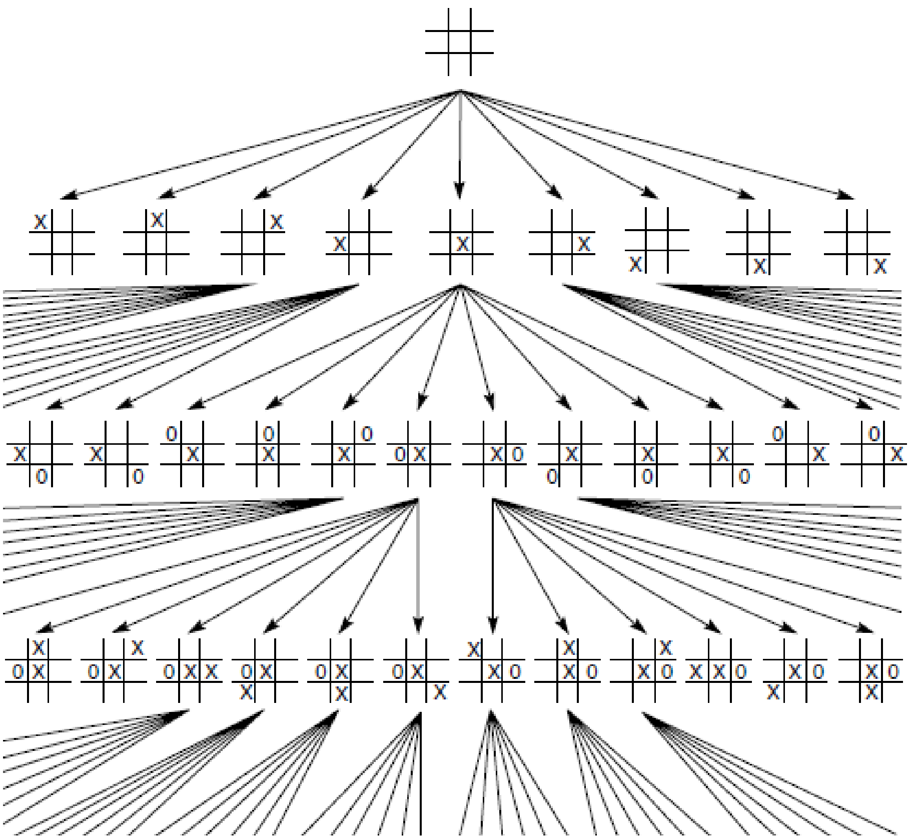
To represent above graph as a tree



From tree we can calculate the cost for each path and determine the minimum one.

Consider, for example, the game of tic-tac-toe. Given any board situation, there is only a finite number of moves that a player can make. Starting with an empty board, the first player may place an X in any one of nine places. Each of these moves yields a different board that will allow the opponent eight possible responses, and so on. We can represent this collection of possible moves and responses by regarding each board configuration as a *node* or *state* in a graph. The *links* of the graph represent legal moves from one board configuration to another. The resulting structure is a *state space graph*.

The state space representation thus enables us to treat all possible games of tic-tac-toe as different paths through the state space graph. Given this representation, an effective game strategy will search through the graph for the paths that lead to the most wins and fewest losses and play in a way that always tries to force the game along one of these optimal paths, as in Figure



Portion of the state space for tic-tac-toe.

كل عقدة في الرسم البياني تمثل "حالة لوحة"

EX: Problem 8-Puzzle:

Initial state

1	4	3
7		6
5	8	2

final or goal state

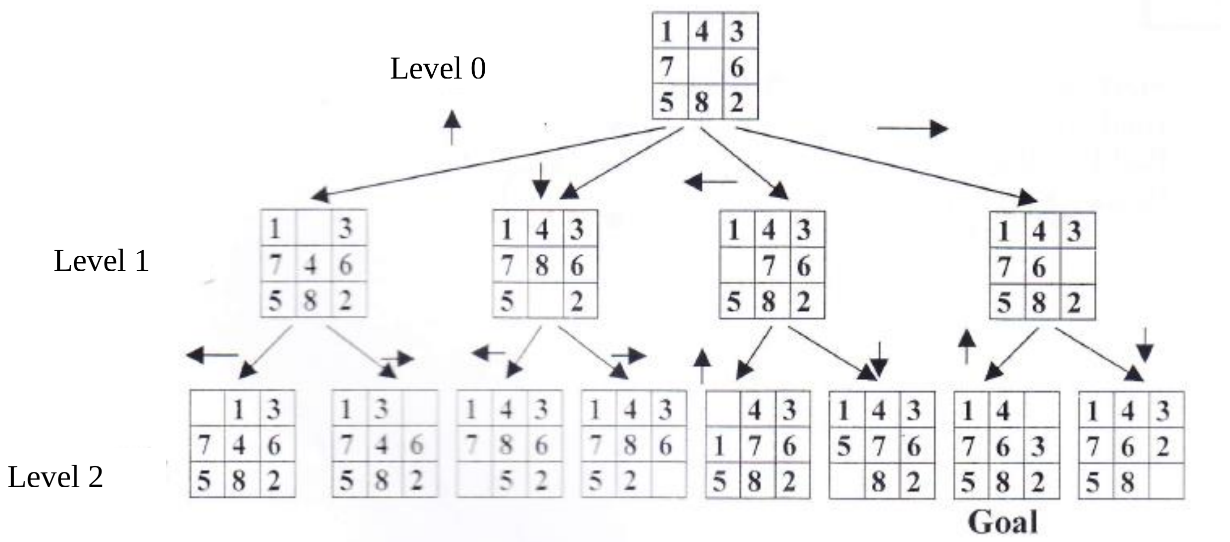
1	4	
7	6	3
5	8	2

Operators : Blank Left ←

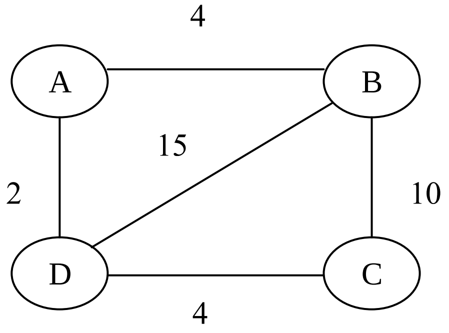
Blank Right →

Blank Up ↑

Blank Down ↓



Example : in the graph shown below, the search algorithm allows you to visit each vertex(node)in the graph in a systematic way.



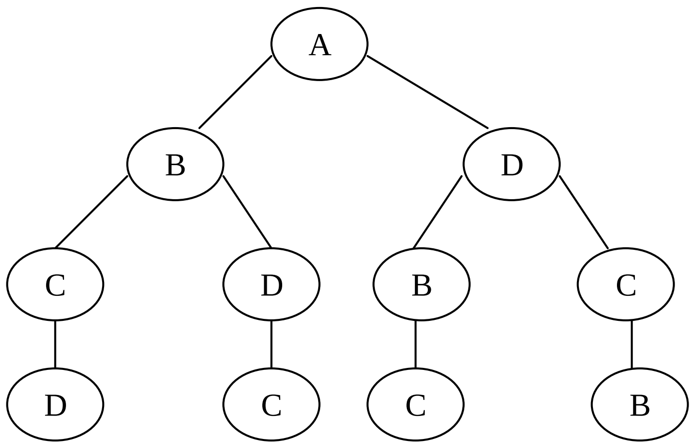
- Convert graph to tree
- Write the paths and calculate
- The cost of each path

A B C D … =4+10+4+2

A B D C …

A D B C …

A D C B …



Number of nods= 11 Links = 10 Paths = 4